Курсовая работа

Тема: Разработка приложения для обмена текстовыми сообщениями по сетям TCP/IP в защищенном исполнении.

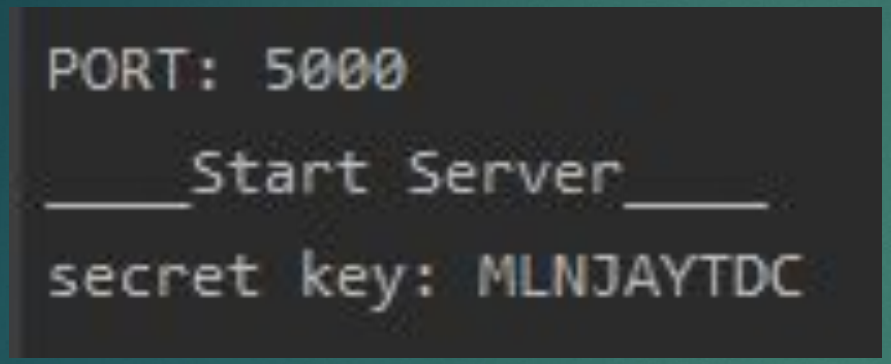
СИЛАЕВА А. О. ИСБ-118

Цели и задачи проекта

- Разработать клиент-серверное приложение для обмена текстовыми сообщениями по сетям TCP/IP
- Реализовать его защиту
- Аудит безопасности приложения одногруппника

Описание работы приложения

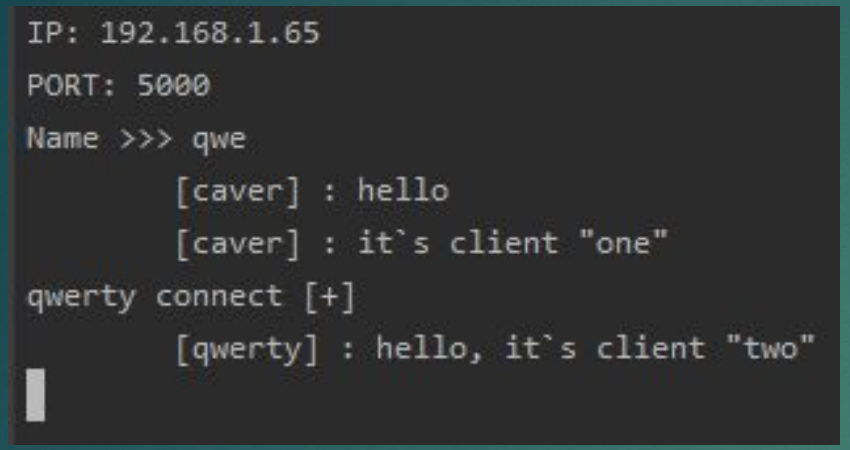
Сервер



При запуске сервера происходит проверка определённых записей в закомментированной области в оригинальном файле “hosts”.
В случае отсутствия этих записей сервер прекращает свою работу

При успешном прохождении проверки, сервер запрашивает порт, после этого, задается кодовое слово для авторизации клиента.

Клиент

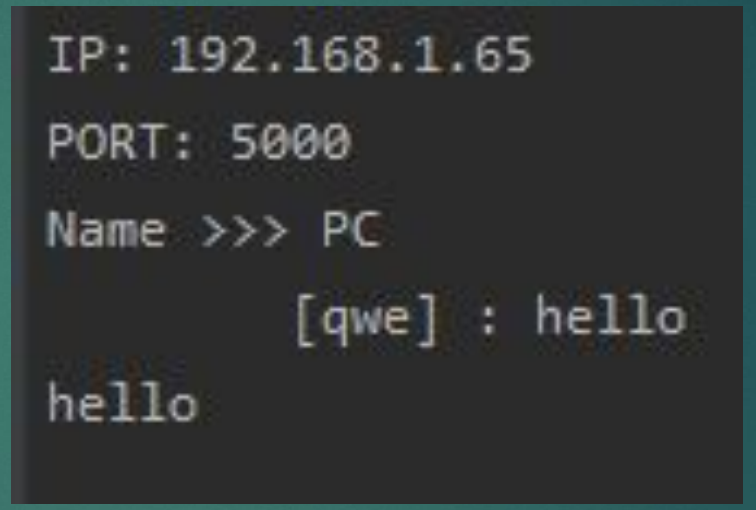


При успешном прохождении проверки, пользователь должен ввести IP, порт, имя и ключ сервера, который генерируется при его запуске. При введении неправильного ключа, пользователь не попадает в чат.

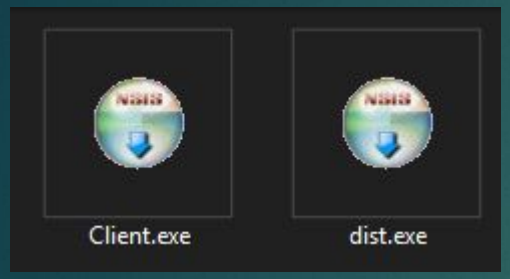
При запске клиента также проходит проверка наличия определённых записей в закомментированной области в оригинальном файле “hosts”. В случае отсутствия этих записей клиент прекращает свою работу

Так же, пользователь сможет увидеть сообщения, которые были отправлены в чате до его появления.

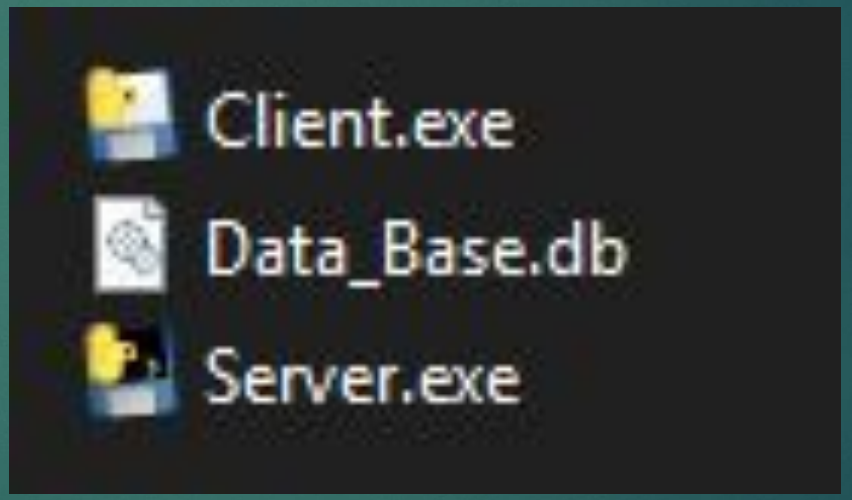
Запуск чата на нескольких устройствах



Взлом оппонента



Изначально, я получил архив, с установочными файлами



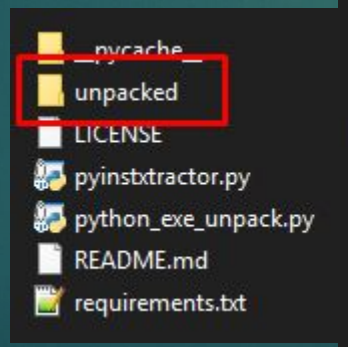
После установки получаем 3 файла. Клиент, сервер и базу данных

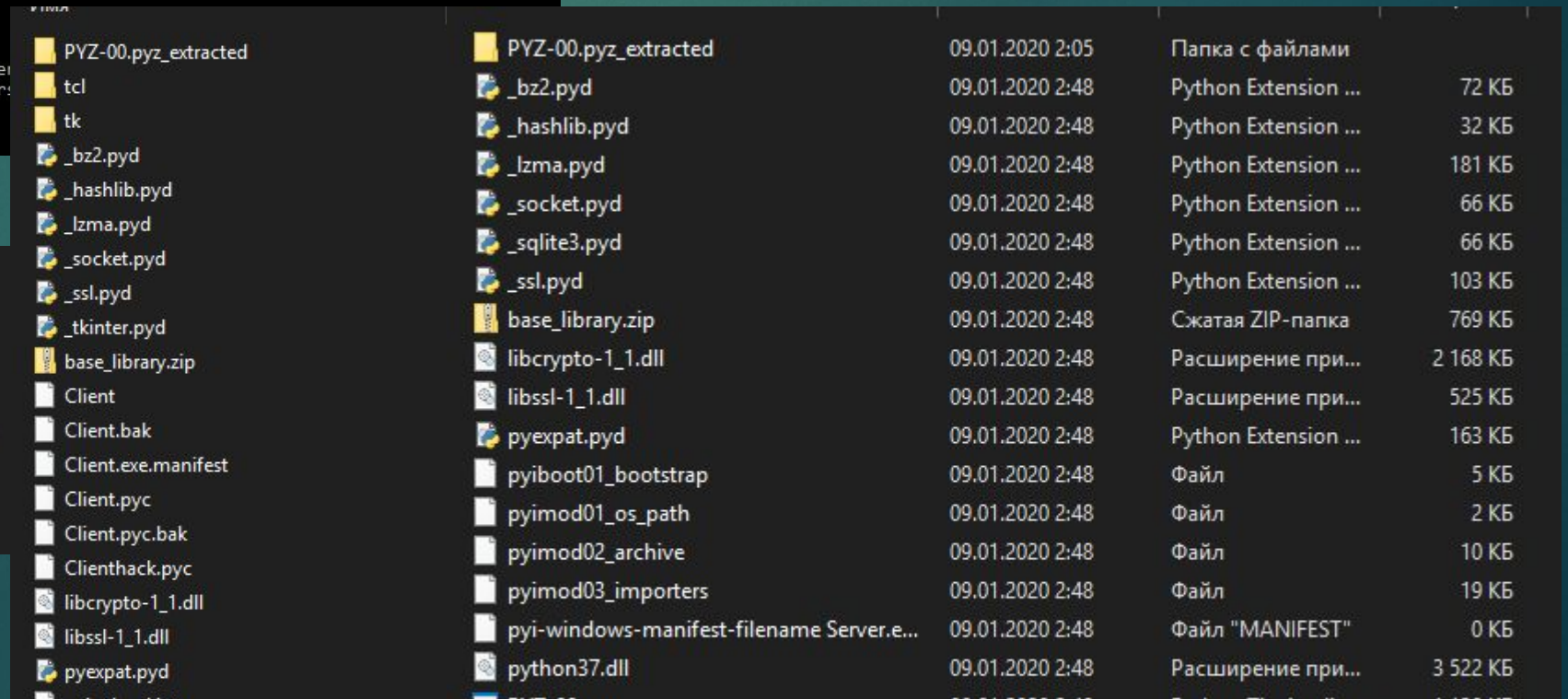
Воспользовавшись утилитой python-exe-unpacker я распаковал .exe файлы

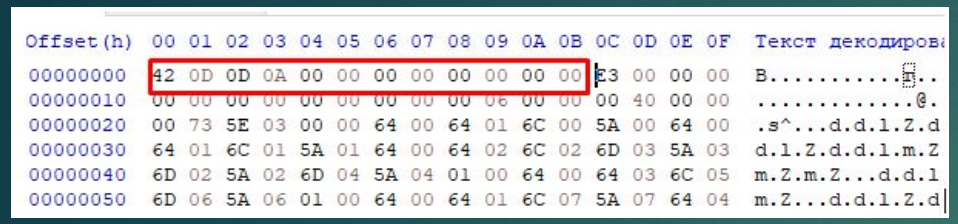
```
C:\Users\Caver\Desktop\dist\server\python-exe-unpacker-master>python_exe_unpack.py -i ..\Server.exe -o unpacked
[*] On Python 3.6
[*] Processing ..\Server.exe
[*] Pyinstaller version: 2.1+
[*] This exe is packed using pyinstaller
[*] Unpacking the binary now
[*] Python version: 37
[*] Length of package: 5337151 bytes
[*] Found 24 files in CArchive
[*] Beginning extraction...please standby
[!] Warning: The script is running in a different python version than the one used to build the executable
    Run this script in Python37 to prevent extraction errors(if any) during unmarshalling
[*] Found 136 files in PYZ archive
[*] Successfully extracted pyinstaller exe.
```

```
C:\Users\Caver\Desktop\dist\client\python-exe-unpacker-master>python_exe_unpack.py -i ..\Client.exe -o unpacked
[*] On Python 3.6
[*] Processing ..\Client.exe
[*] Pyinstaller version: 2.1+
[*] This exe is packed using pyinstaller
[*] Unpacking the binary now
[*] Python version: 37
[*] Length of package: 7195067 bytes
[*] Found 938 files in CArchive
[*] Beginning extraction...please standby
[!] Warning: The script is running in a different python ver
    Run this script in Python37 to prevent extraction errors
[*] Found 135 files in PYZ archive
[*] Successfully extracted pyinstaller exe.
```

и получил файлы







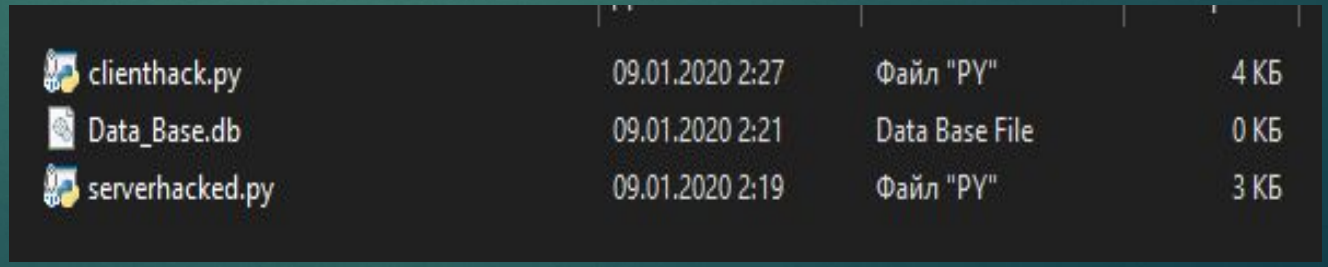
Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Текст декодирова
00000000	42	0D	0D	0A	00	00	00	00	00	00	00	00	E3	00	00	00	B.....ã..
00000010	00	00	00	00	00	00	00	00	00	06	00	00	00	40	00	00	.....@.
00000020	00	73	5E	03	00	00	64	00	64	01	6C	00	5A	00	64	00	.s^...d.d.l.Z.d
00000030	64	01	6C	01	5A	01	64	00	64	02	6C	02	6D	03	5A	03	d.l.Z.d.d.l.m.Z
00000040	6D	02	5A	02	6D	04	5A	04	01	00	64	00	64	03	6C	05	m.Z.m.Z...d.d.l
00000050	6D	06	5A	06	01	00	64	00	64	01	6C	07	5A	07	64	04	m.Z...d.d.l.Z.d

Воспользовавшись Hex-редактором, я получил файл .pyc и добавил заголовки, которые использовались другими .pyc файлами

```
C:\Users\Caver\Desktop\dist\client\python-exe-unpacker-master\unpacked\Client.exe>uncompyle6 Clie
nt.pyc
# uncompyle6 version 3.6.2
# Python bytecode 3.7 (3392)
# Decompiled from: Python 3.6.0 (v3.6.0:41df79263a11, Dec 23 2016, 07:18:10) [MSC v.1900 32 bit (
Intel)]
# Embedded file name: Client.py
import hashlib, sys
```

После этого, я декомпилировал файлы .pyc с помощью утилиты uncompyle6

И получил исходные файлы



C:\Users\Caver\Desktop\dist\done\clienthack.py (done) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

FOLDERS

done

clienthack.py

serverhacked.py

serverhacked.py | clienthack.py

```python
import hashlib, sys
from socket import AF_INET, socket, SOCK_STREAM
from threading import Thread
import tkinter

def quit():
    soket.send(bytes('/exit', 'utf8'))
    sys.exit()


def log(event=None):
    nickname = nic.get()
    password = pas.get()
    h = hashlib.md5(password.encode())
    ht = h.hexdigest()
    soket.send(bytes(nickname, 'utf8'))
    soket.send(bytes(ht, 'utf8'))
    msg = soket.recv(buff).decode('utf8')
    if msg == '1':
        login.destroy()
    else:
        label = tkinter.Label(login, text='Ошибка входа.', fg='red')
        label.place(x=20, y=90)


def conn(event=None):
    global HOST
    global PORT
    HOST = host.get()
    PORT = port.get()
    connect.destroy()


def send(event=None):
    msg = my_msg.get()
    my_msg.set('')
    soket.send(bytes(msg, 'utf8'))


def read_sock(event=None):
    while True:
        msg = soket.recv(buff).decode('utf8')
        msg_list.insert(tkinter.END, msg)


connect = tkinter.Tk()
connect.title('Подключение')
connect.geometry('300x130')
connect.resizable(width=False, height=False)
host, port = tkinter.StringVar(), tkinter.StringVar()
label1 = tkinter.Label(connect, text='Хост:')
label2 = tkinter.Label(connect, text='Порт:')
entry1 = tkinter.Entry(connect, textvariable=host)
entry2 = tkinter.Entry(connect, textvariable=port)
button = tkinter.Button(connect, text='Отправить')
button.bind('<Button-1>', conn)
label1.place(x=20, y=20)
label2.place(x=22, y=50)
```

Line 105, Column 16 Spaces: 4 Python

C:\Users\Caver\Desktop\dist\done\serverhacked.py (done) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

FOLDERS

done

clienthack.py

serverhacked.py

serverhacked.py | clienthack.py

```python
def process_conn():
    while True:
        client, client_address = SERVER.accept()
        name = login(client)
        print('%s:%s подключился.' % client_address)
        addresses[client] = client_address
        Thread(target=processing_client, args=(client, name)).start()


def processing_client(client, name):
    clients[client] = name
    send(bytes('%s присоединился' % name, 'utf8'))
    while True:
        msg = client.recv(buff)
        if msg == bytes('/exit', 'utf8'):
            print('%s:%s отключился' % addresses[client])
            client.close()
            del clients[client]
            break
        elif msg == bytes('/register', 'utf8'):
            if name != 'Admin':
                client.send(bytes('Нет доступа', 'utf8'))
            else:
                connection = sqlite3.connect('Data_Base.db')
                cursor = connection.cursor()
                client.send(bytes('Введите никнейм', 'utf8'))
                nickname = client.recv(buff).decode('utf8')
                client.send(bytes('Введите пароль', 'utf8'))
                password = client.recv(buff).decode('utf8')
                h = hashlib.md5(password.encode())
                ht = h.hexdigest()
                zapros = "INSERT INTO Login(Nickname, Password) VALUES('{}','{}')".format(nickname, ht)
                cursor.execute(zapros)
                connection.commit()
                connection.close()
                client.send(bytes('Пользователь зарегистрирован', 'utf8'))
        else:
            send(msg, name + ': ')


def send(msg, prefix=''):
    for sock in clients:
        sock.send(bytes(prefix, 'utf8') + msg)


clients = {}
addresses = {}
HOST = socket.gethostbyname(socket.gethostname())
print('Хост: ' + HOST)
PORT = 9090
print('Порт: ' + str(PORT))
buff = 1024
ADDR = (HOST, PORT)
SERVER = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
SERVER.bind(ADDR)
SERVER.listen(10)
print('Сервер запущен.')
```

Line 24, Column 16 Spaces: 4 Python

Конец.